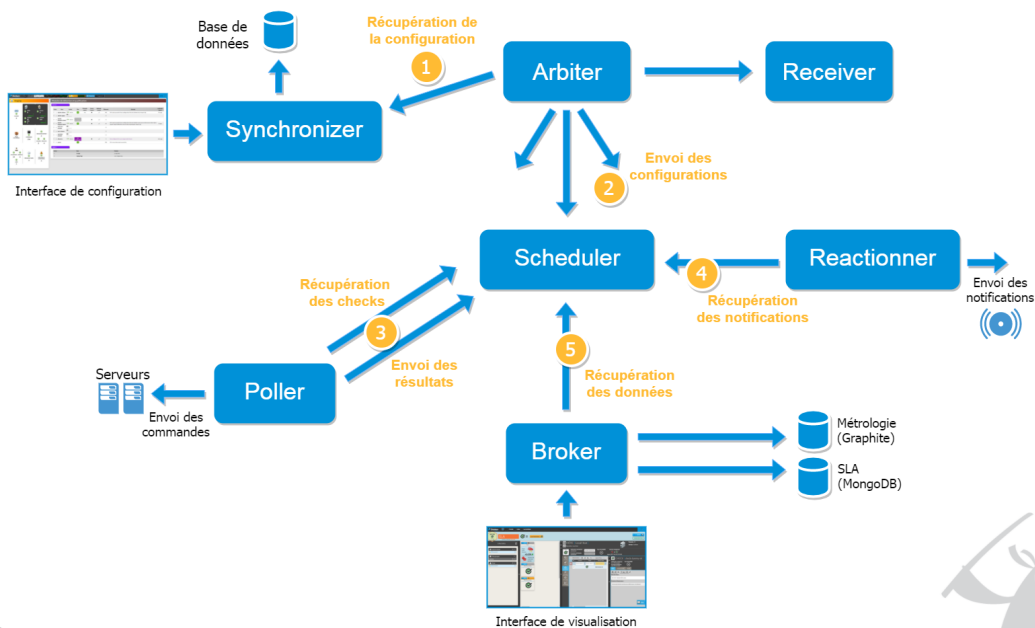


# Architecture d'une installation Shinken Enterprise

## Principes généraux

L'architecture est totalement flexible et évolutive (scalable). Pour améliorer les capacités de Shinken Enterprise, augmenter le nombre de démons ayant le même rôle est la meilleure approche.

### Architecture Globale



## Répartition de charge automatique (load balancing)

### Distribution des hôtes à travers les schedulers

Shinken Enterprise, via le démon Arbiter, est capable de découper la configuration en plusieurs parties et les distribuer aux Scheduler.

- Le load balancing est fait automatiquement, l'administrateur n'a pas besoin de se souvenir quel hôte est lié à tel autre pour créer les packs.
- La répartition est basée sur les hôtes : cela veut dire que tous les checks associés à un hôte seront dans le même Scheduler que l'hôte. Cela signifie que l'administrateur n'a pas besoin de connaître toutes les relations entre éléments comme les parents, dépendances d'hôtes ou dépendances de checks. Shinken Enterprise est capable de lire ses relations et de rassembler tous les éléments liés dans la même partition.

Cette action se fait en 2 parties :

- Création de partitions indépendantes pour les éléments
- Copie des partitions pour créer N configurations pour N Schedulers

## Création de partitions indépendantes

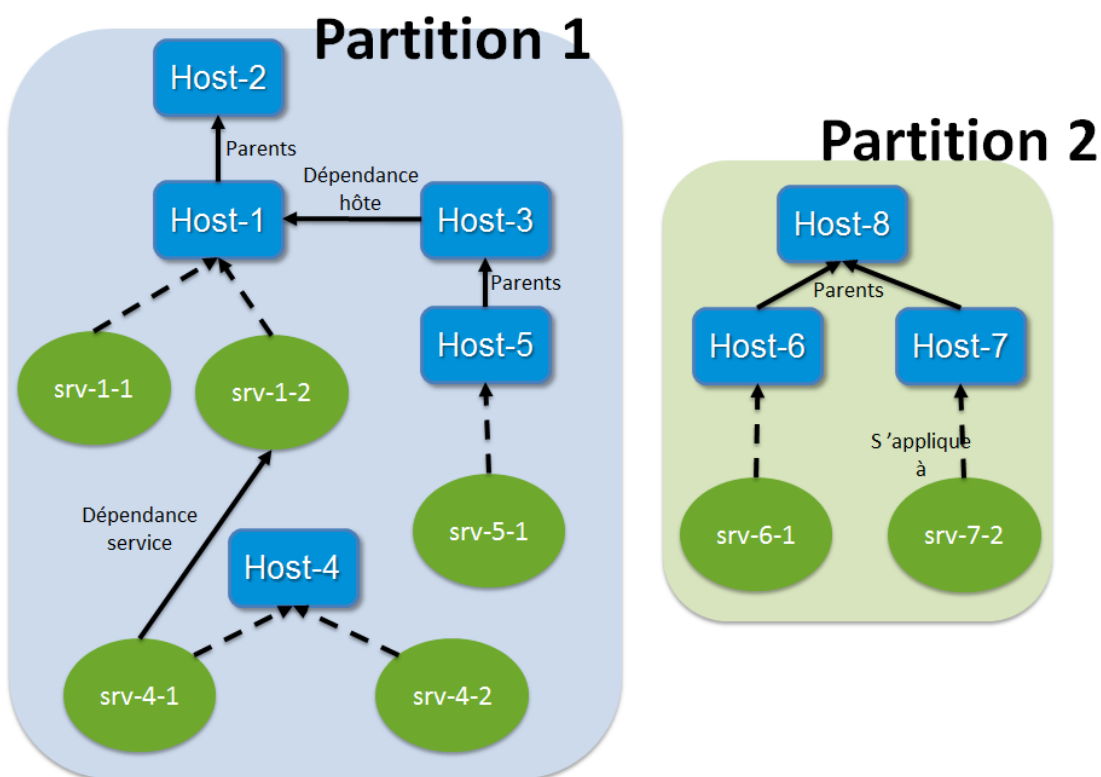
L'action de hachage se fait en se basant sur 2 éléments : les hôtes et les checks. Les checks sont liés à l'hôte donc ils seront dans la même partition.

D'autres relations sont prises en compte :

- Liaisons réseau pour un hôte (comme un serveur distant et son routeur).
- Dépendances logiques.

Shinken Enterprise regarde toutes les relations et crée un graphe avec. Un graphe est une partition de relations.

Illustration :



Dans cet exemple, nous avons 2 partitions:

- Shard 1: Host-1 au host-5 et tous leurs checks
- Shard 2: Host-6 au Host-8 et tous leurs checks

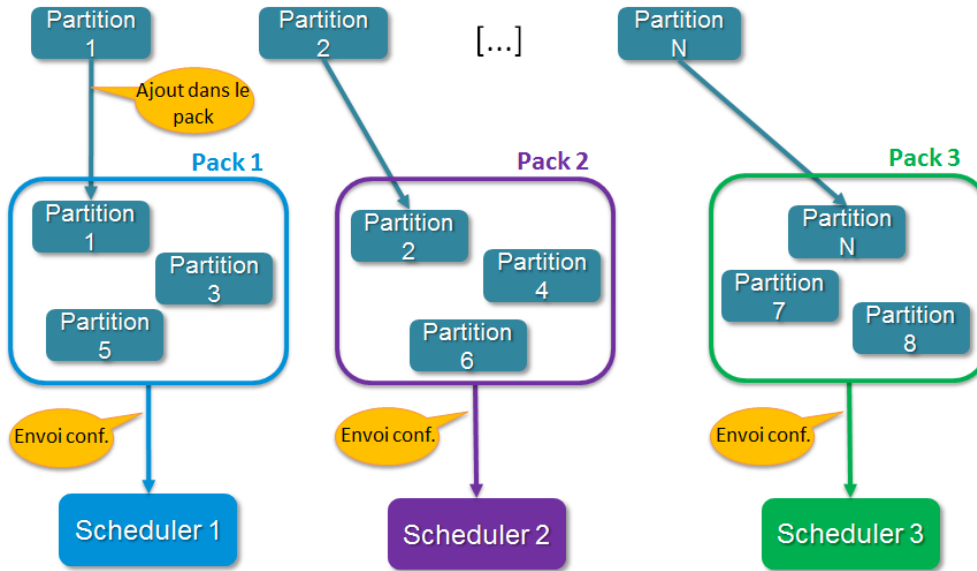
## L'agrégation des partitions dans les schedulers

Quand toutes les partitions sont créées, l'Arbiter les agrège dans N configurations si l'administrateur a défini N Schedulers actifs (sans spare).

Pour rappel, un spare est un élément qui prendra le relais si le nœud actif venait à s'arrêter.

La répartition se fait sur un critère de poids des Schedulers : plus le poids est élevé, plus il y a de packs.

Illustration :



## Envoi des configurations vers des satellites

Une fois que toutes les configurations sont créées, l'Arbiter les envoie aux N Schedulers actifs .

Un Scheduler peut commencer à lancer des checks une fois qu'il a reçu et chargé sa configuration sans avoir à attendre que TOUS les Schedulers soient prêts.

Pour des configurations plus importantes, avoir plusieurs Schedulers (même sur un seul serveur) est fortement recommandé car ils chargeront leur configuration beaucoup plus vite (nouvelle ou modification).

L'Arbiter crée également les configurations pour ses satellites (pollers, reactionners et brokers) avec les liens permettant de savoir où réaliser les tâches.

Il est également responsable de vérifier la disponibilité des satellites.

## La haute disponibilité

L'architecture de Shinken Enterprise est hautement disponible.

Un serveur peut crasher, une application également. C'est pour cela que les administrateurs ont des backup : ils peuvent recharger la configuration des éléments tombés.

L'Arbiter vérifie régulièrement que tous les démons sont disponibles. Si un nœud n'est plus accessible, l'Arbiter envoie sa configuration au nœud spare défini par l'administrateur (s'il existe).

- Tous les satellites sont informés de ce changement, de façon à recevoir leur tâches du nouvel élément sans essayer de joindre le démon tombé.
- Si un nœud est perdu à cause d'une coupure réseau, puis revient, l'Arbiter en prend note et demande au système de supprimer son ancienne configuration. Le démon maître reprend le relais.

Une page de documentation est dédiée à la Haute disponibilité [ici](#).

## Distribution par Commandes Externes

Après avoir envoyé les configurations et les partitions, l'Arbiter commence à traiter les ordres, appelées commandes externes, qui sont des commandes qui vont s'exécuter via les actions faites par l'utilisateur de l'interface de visualisation mais également lors de la réception de [check passif](#) que le démon Receiver récupère.

De cette manière, des ordres peuvent être envoyés aux Schedulers, via l'Arbiter.

Les commandes externes sont de 2 types :

- commandes qui concernent tous les Schedulers.
- commandes qui sont spécifiques à un seul élément (hôte/check).

Pour chaque commande, Shinken Enterprise détecte si c'est global ou particulier:

- Si global, il envoie les ordres à tous les Schedulers.
- Si particulier, il détecte quel Scheduler gère l'élément concerné par la commande (hôte/check) et envoie l'ordre au bon Scheduler.

Dès réception de l'ordre par le ou les Schedulers, il est appliqué.